Airbert: In-domain Pretraining for Vision-and-Language Navigation
SUPPLEMENTARY MATERIAL

Pierre-Louis Guhur[1], Makarand Tapaswi[2], Shizhe Chen[1], Ivan Laptev[1], Cordelia Schmid[1]
[1]Inria, École normale supérieure, CNRS, PSL Research University, Paris, France
[2]IIIT Hyderabad, India
pierre-louis.guhur@inria.fr https://airbert-vln.github.io

In this supplementary material, we present additional details, statistics and examples for the BnB dataset; we discuss implementation details for the models used in our work; and present qualitative results as well as the detailed results for the new few-shot learning paradigm.

A. BnB dataset

This section presents additional details for our *Bed-and-Breakfast* (BnB) dataset. We start by a short discussion of image-caption pairs (BnB IC) collected from an online rental marketplaces and their statistics. Subsequently, we present how a combinatorially large number of path-instruction pairs (BnB PI) can be created automatically. We end this section with multiple examples of BnB PI pairs generated via the concatenation and domain-shift reduction (*e.g.* rephrasing, captionless insertion) strategies.

A.1. Filtering image-caption pairs: Outdoor images

Images of outdoor scenes are almost never seen in the environments used in downstream VLN tasks. In fact, not only are the images out-of-domain (such images are rarely seen in the VLN environments), their captions are often irrelevant to a VLN task. In order to alleviate the impact of such noisy images and captions, we discard outdoor images from the pretraining process. Figure 1 illustrates several examples of misleading outdoor image-caption pairs. Captions as written by the host are presented in the label below the image. The caption for the image in Figure 1a refers to a "*bedroom*", however, the image does not show a bedroom. Similarly, the image-caption pair in the Figure 1b talks about activities or festivals that take place in the neighborhood of the listing, however, they are not relevant for solving indoor navigation tasks. Finally, Figure 1c shows an outdoor scene with several birds along with a noisy caption that is not directly related to the image content, but the emotion that the image may evoke.

A.2. Dataset details and Statistics

BnB image-caption pairs. We collect BnB IC pairs from 150K listings on *Airbnb* resulting in 713K image-caption pairs and 676K images without captions. In Figure 2, we present some key statistics about this data. Figure 2a presents a histogram of the number of images found in each listing. While most listings have less than 20 images, this is still a sufficiently large and diverse in-domain distribution. In Figure 2b, we summarize the rooms depicted in the images through predicted category labels obtained using a CNN trained on the *Places365* dataset [13]. These category labels are used as part of our proposed extensions such as *image merging*.

Creating BnB path-instruction pretraining samples. We create the BnB PI pairs on-the-fly during training to mimic the agent's visual trajectory and a corresponding instruction through an environment. Each sample in a batch is created by randomly sampling a listing without replacement during an epoch (one epoch consists of one PI pair from each listing). Then, the number of IC pairs K that form the PI pair are chosen (as an integer) from a uniform distribution, $K \sim U[4, 7]$. We sample $N \sim U[2, K]$ IC pairs that have a non-empty caption and the remainder $K - N$ images are chosen from the set of captionless images. Any image in the path may include additional visual context (from the same room) via the *image merging* strategy. Similarly, the *instruction rephrasing* strategy may be employed by using existing R2R instruction templates and filling them with noun phrases extracted from the image captions.

The above procedure results in creating one correctly aligned (positive) PI pair, (X^+ in the main paper). To employ the shuffling loss for each sample, we create 9 additional negatives (X_n^- in the paper) by shuffling either the sequence of images or captions, ensuring that the post-shuffling order does not align with the positive pair.

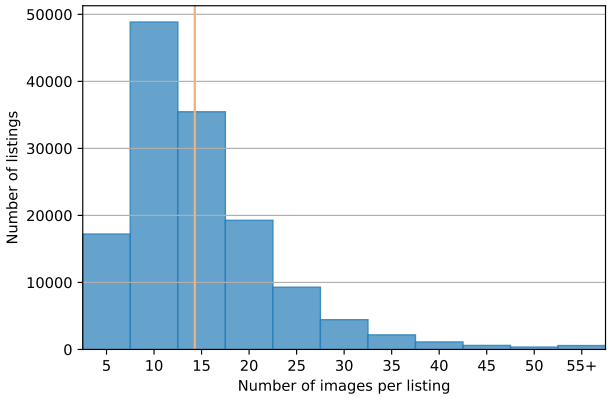
Statistics for BnB PI pairs. Due to the large number of possible combinations, we can (theoretically) create 200

(a) Gorgeous ocean views from bedroom.

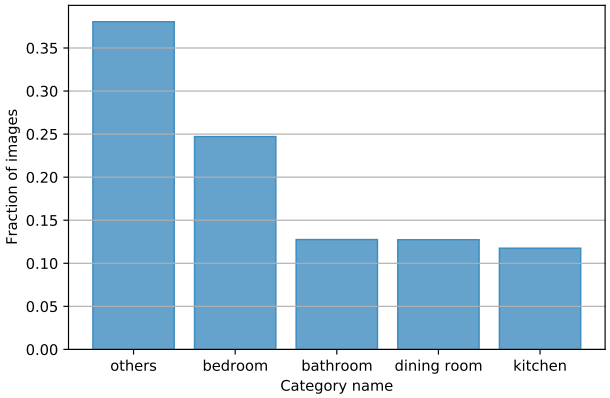
(b) Main stage

(c) Spring excitement

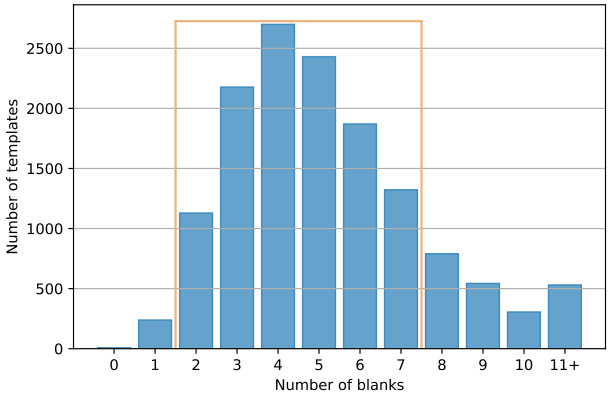
Figure 1: Examples of outdoor images with their corresponding captions.



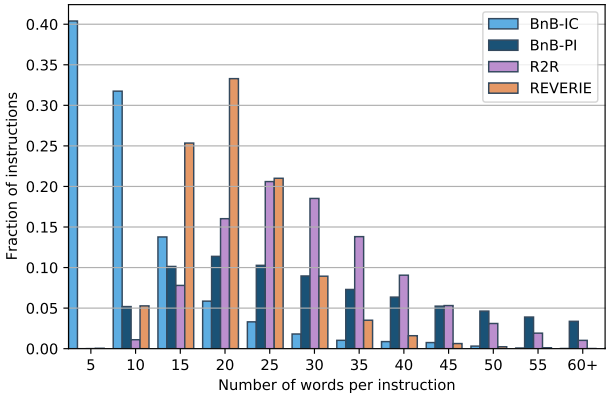
(a) Distribution of the number of images per listing.



(b) Distribution of predicted scene categories on BnB images.



(c) Fill-in-the-blanks templates built using the R2R training set.



(d) Distribution of the instruction lengths.

Figure 2: Statistics of BnB Dataset.

billion path-instruction pairs, using the simple concatenation strategy. This number grows to over 300 quadrillion when considering additional visual context augmentations and fluent instructions.

For *instruction rephrasing*, we create 11,626 fill-in-the-blank templates from the R2R training set. Figure 2c shows the distribution of the number of blanks in the templates – most instruction templates have 2-7 blanks into which we insert noun phrases from the BnB captions.

While we are unable to generate the entire BnB PI dataset for computing statistics, we generate 50K PI pairs as a representative sample. Figure 2d presents the distribution of instruction lengths (number of words) for different datasets. We see that the captions in BnB IC pairs are much shorter than typical instructions in R2R and REVERIE, while our automatically created instructions in BnB PI pairs

exhibit a high level of similarity in terms of their length.

A.3. Examples of BnB PI Pairs

Figure 3 presents generated BnB PI pairs using various strategies proposed in our work, including naive concatenation, instruction rephrasing, instruction generation, image merging and captionless image insertion.

Among the methods to create an instruction, simple concatenation lacks action verbs between sentences for fluent transition leading to a domain shift from real instructions. Instruction rephrasing selects noun phrases from BnB image descriptions and inserts them into real instruction templates, providing a natural feel to the created instruction. Finally, while the learning approach of instruction generation (recall, this is learned on downstream VLN dataset) produces fluent sentences, it is unable to leverage the diverse captions of BnB images due to the limited vocabulary stemming from the downstream VLN dataset. For example, the generated instruction in Figure 3c does not contain noun phrases related to images in the path. Better caption generation models such as Pointer network [12] may help avoid such problems, however are left for future work.

Among augmentations for path generation, we can see that *image merging* helps to expand relevant visual context from single images to semi-panoramic views, see the bedroom in Figure 3a or the kitchen in Figure 3b. *Captionless image insertion* also improves the path diversity by mimicking unmentioned viewpoints in the instruction (indicated by images with a dotted border).

B. Implementation details

We present the implementation details for learning Airbert via pretraining using BnB, and subsequent fine-tuning in both discriminative or generative settings.

B.1. Airbert Pretraining

Airbert's architecture is the same as VLN-BERT (see Figure 4a where the number of layers $L_1 = L_2 = 6$). The feature vector v_i^k (corresponding to ith image region of the kth image) is composed of three terms: the first term is the visual feature extracted by the Bottom-Up Top-Down attention model [1]; the second term encodes the location of the region in the image as MLP(l_i^k), where l_i^k is the 5-dim location vector of the given image region defined as the top corner (x, y), the width, height and area; and the last term Emb(k) encodes the position, where Emb is an embedding layer for the image order.

We use 8 V100 SXM2 GPUs (32 GB each) for pretraining Airbert. The model is trained for 15 epochs with a batch size of 64 and learning rate of 4×10^{-5}. Each epoch consists of one randomly sampled PI pair from 95% of the listings, while the remaining 5% are used for validation and preventing overfitting.

B.2. Fine-tuning in Discriminative Setting

In the discriminative setting, R2R navigation is formulated as a path selection problem given the instruction. The pretrained Airbert model can be directly fine-tuned without any modifications to the architecture to predict the path-instruction alignment (or compatibility) score as shown in Figure 4a.

We follow the same fine-tuning setup as VLN-BERT [9] to allow for a fair comparison. We use the Adam optimizer with a learning rate of 4×10^{-5}. The optimizer is controlled by a learning rate scheduler with a linear warmup and cooldown. We fine-tune Airbert for 30 epochs with a batch size of 64. Samples from the R2R training set are used for fine-tuning and the model checkpoint with the highest success rate on the unseen validation set (val unseen) is selected for the test set and leaderboard submission.

B.3. Fine-tuning in Generative Setting

In the generative setting, an agent is required to predict navigable actions step by step. We adopt the state-of-the-art generative model Recurrent VLN-BERT [6] for R2R and REVERIE tasks. The model uses a pretrained multimodal transformer as a backbone and adds recurrence to a state token to keep track of history for sequential action prediction. Although the original Recurrent VLN-BERT model only implements an LXMERT-like [10] architecture PREVALENT [5], and one-stream BERT-like architecture OSCAR [8], it is easy to plug our two-stream ViLBERT architecture as the backbone.

The adapted model is shown in Figure 4b. For initialization, the language stream is used to encode the instruction C into an instruction representation H. As no visual inputs are used during the initialization, the co-attention modules in the original language stream of ViLBERT are removed, and the output feature of the `[CLS]` token is used as the agent's initial state s_0. For navigation at each step k, the visual stream takes the previous state s_{k-1}, visual observations V_k at step k and the encoded language features H to generate a new state s_k and action decision p_k.

When fine-tuning on the R2R dataset, we use scene features with a ResNet-152 pretrained on Places365 [13] and augment the training data with generated path-instruction pairs from [5]. We train the model via imitation learning and A2C reinforcement learning for 300,000 iterations with a batch size of 16 and learning rate of 10^{-5}. When fine-tuning on the REVERIE dataset, object features encoded by a Bottom-Up Top-Down attention model [1] are used along with the scene features. The model is trained for 200,000 iterations with a batch size of 8. All the experimental setups for fine-tuning are the same as [6] for a fair comparison.

Concatenation: extra guest room with comfy full bed on top floor of house and top floor shared bathroom for both guest rooms then adjoining modern private bath with stall shower bath and beach towels provided then granny's treasures add a homey touch
Instruction rephrasing: exit extra guest room and turn left. pass top floor shared bathroom then turn right. walk toward a homey touch and wait there.
Instruction generation: walk to the other side of the bathroom and stop next to the last corner on the wall with the candles.

(a) Example 1

Concatenation: full bath and open floor plan living opens to deck, kitchen / dining area
Instruction rephrasing: go around full bath, then open floor plan living down to kitchen / dining area.
Instruction generation: walk into the bathroom and turn right. walk to the end of the landing and turn left. walk into the sitting area and turn right. walk past the chair and stop.

(b) Example 2

Concatenation: bedroom 3 (picture 2 of 2) - 2 twin beds w / full size washer then bedroom 2 - queen bed - 1st floor, bathroom 1st floor. w / tub and shower.
Instruction rephrasing: exit the bedroom 3 and go right into bedroom 2 next to tub and shower.
Instruction generation: walk straight through the doorway and turn right. walk straight through the doorway and turn left. walk through the doorway and stop.

(c) Example 3

Figure 3: Examples of path-instruction pairs created by different strategies. The images with dotted borders are images chosen from the *captionless image insertion* strategy, and the clustered images are from the *image merging* strategy.

C. Results

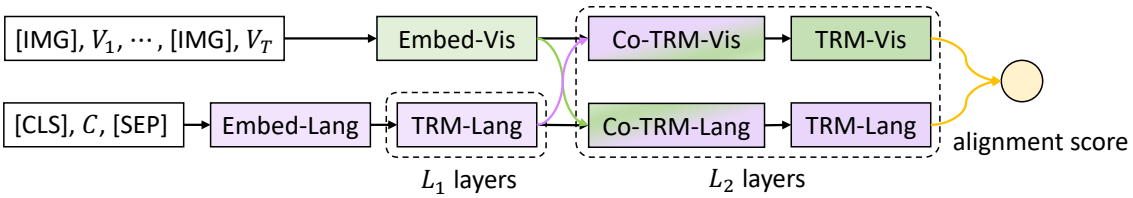
In this section, we present additional results on adapting Airbert to a generative setting and applying it to the R2R task. Through several qualitative examples, we obtain a better understanding for Airbert's performance improvements, and finally present detailed results on the new few-shot learning paradigm in VLN.

C.1. Results on R2R with Generative Models

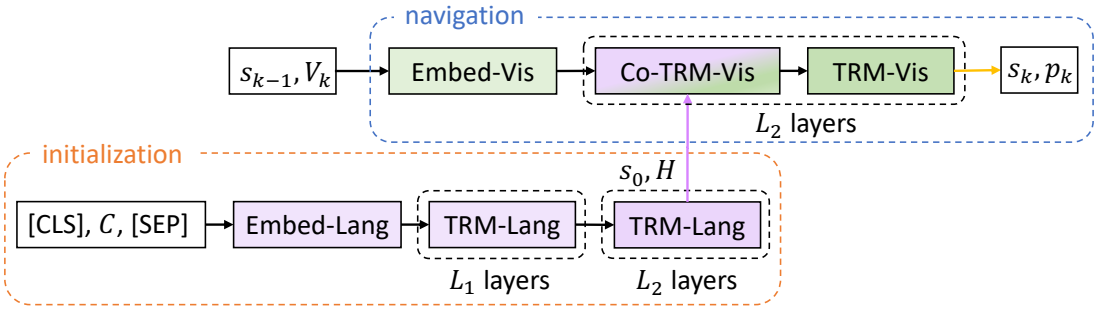
Table 1 shows the performance of different generative models on the R2R dataset. The OSCAR and ViLBERT backbones for Recurrent VLN-BERT [6] (Rec) are all pretrained on large-scale out-of-domain image-caption pairs with object features and similar self-supervised tasks. On the other hand, the PREVALENT [5] backbone is pretrained on in-domain R2R dataset with scene features and fine-tuned with an additional action prediction task. We suspect that this is the reason for PREVALENT's higher performance as compared to using OSCAR or VLN-BERT as backbones. Note that our Airbert backbone is not fine-tuned further on downstream tasks after pretraining.

Replacing OSCAR's single BERT-like architecture with the ViLBERT architecture slightly improves the performance (similar to our results on the REVERIE dataset presented in the main paper). The VLN-BERT model further fine-tunes ViLBERT on the R2R dataset (with the masking loss). This is beneficial to the navigation performance on the unseen environments validation set[1]. Our Airbert initialization achieves substantial performance improvement as compared to the OSCAR and VLN-BERT backbones on unseen environments, while achieving comparable performance with the PREVALENT initialization.

[1]The performance of VLN-BERT on the seen validation set is lower because the model checkpoint is selected to maximize performance on validation unseen set which happens to be at an earlier iteration.



(a) Adapting Airbert to a discriminative setting to predict path-instruction alignment score, similar to [9].



(b) Adapting Airbert to a generative setting based on the Recurrent VLN-BERT [6].

Figure 4: The adapted Airbert model in both discriminative and generative settings for downstream VLN tasks.

Methods	Validation Seen				Validation Unseen				Test Unseen			
	TL	NE	SR	SPL	TL	NE	SR	SPL	TL	NE	SR	SPL
Seq2Seq-SF [2]	11.33	6.01	39	-	8.39	7.81	22	-	8.13	7.85	20	18
Speaker-Follower [4]	-	3.36	66	-	-	6.62	35	-	14.82	6.62	35	28
PRESS [7]	10.57	4.39	58	55	10.36	5.28	49	45	10.77	5.49	49	45
EnvDrop [11]	11.00	3.99	62	59	10.70	5.22	52	48	11.66	5.23	51	47
PREVALENT [5]	10.32	3.67	69	65	10.19	4.71	58	53	10.51	5.30	54	51
Rec (no init. OSCAR) [6]	9.78	3.92	62	59	10.31	5.10	50	46	11.15	5.45	51	47
Rec (OSCAR) [6]	10.79	3.11	71	67	11.86	4.29	59	53	12.34	4.59	57	53
Rec (PREVALENT) [6]	11.13	2.90	72	68	12.01	3.93	63	57	12.35	4.09	63	57
Rec (ViLBERT)	11.16	2.54	75	71	12.44	4.20	60	54	-	-	-	-
Rec (VLN-BERT)	10.95	3.37	68	64	11.33	4.19	60	55	-	-	-	-
Rec (Airbert)	11.09	2.68	75	70	11.78	4.01	62	56	12.41	4.13	62	57

Table 1: Navigation performance of different generative models on the R2R dataset.

C.2. Qualitative results

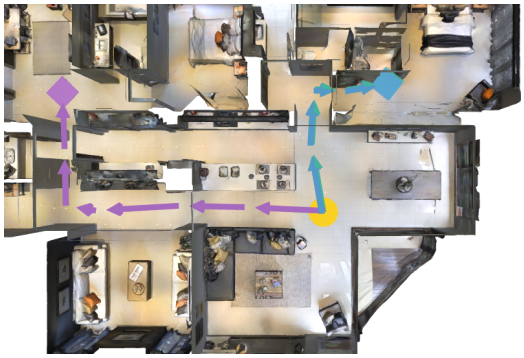
We visualize the predicted paths from VLN-BERT and Airbert models. In the following figures, ● is the starting viewpoint of the agent, ■ denotes viewpoints in the ground-truth path, ■ for VLN-BERT and ■ for Airbert. Arrows indicate the navigation direction.

New houses. In Figure 5, we compare predicted paths from VLN-BERT and Airbert in new houses beyond the training environments. Benefiting from BnB dataset that provides diverse visual environments in pretraining, our Airbert model generalizes better to recognize different room types in new houses (see Figure 5a-5d), and performs better on significantly different environments such as a church (Figure 5e) or castle (Figure 5f).

New objects. Airbert also improves the understanding of new objects in home environments, *e.g.* through noun phrases related to household objects. As shown in Figure 6, it is successful at following instructions containing noun phrases that rarely occur or are even unseen on the training set, while the VLN-BERT model that is trained on a large image-caption corpus not pertaining to houses fails.

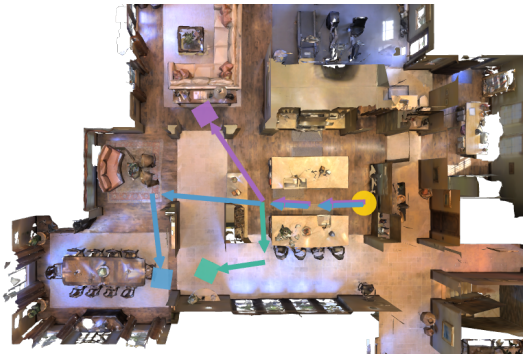
Similar environments and instructions. Figure 7 displays examples where the environments and the instructions are similar to those on the training set, with the aim to show that the shuffling loss in pretraining also benefits learning.

(a) **R2R ✔:** Walk over the kitchen counter, turn left, walk ahead till wall, turn right, walk to the closet room, wait at front.

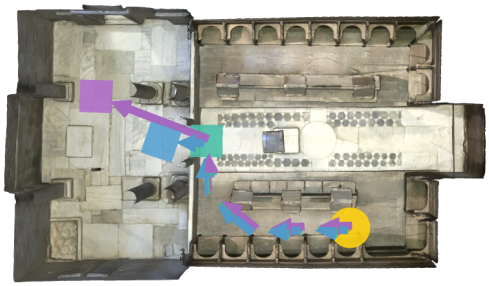


(b) **REVERIE ✔:** Walk past the kitchen and enter the hallway. Turn right at the artwork and wait by the closet.

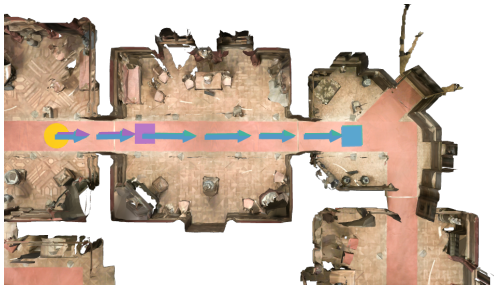
(c) **R2R ✔:** Walk forward to the sitting area to the right of the stairs. Walk to the wall of windows and take a right into the recreation room and stop before you reach the pool table.



(d) **R2R ✔:** Go between the counters, turn left, turn right, and stop before the display and dining room.



(e) **R2R ✔:** Turn right and head towards the end. Once you reach the end make a right and stop.

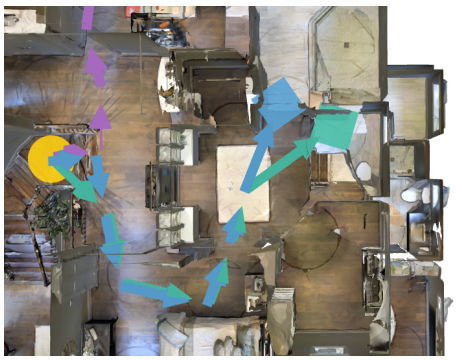


(f) **R2R ✔:** Walk straight out the door in front of you and follow the red carpet. Keep going through the room with the ropes and stop when you enter the next room with ropes.

Figure 5: When navigating in new houses, our Airbert model not only successfully recognizes the closet room in (a) and (b), pool table (c), living room (d), but also generalizes better to challenging environments, such as the church (e) and castle (f).

For example, in Figure 7a, the VLN-BERT agent ■ focuses on the stairs (in the last step) and goes upstairs incorrectly, whereas Airbert learns to consider intermediate steps such as "*lounge chairs*" and "*cabinet*" besides the last step by learning from the shuffling task. Similarly, in Figure 7c, we see that the VLN-BERT agent stops at the wrong stairs, while Airbert considers intermediate steps such as "*hallway*" and "*wooden doors*", and ends within the acceptable range of 3m from the goal.

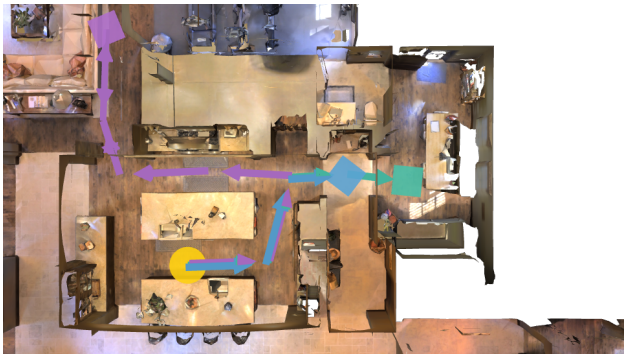
Failure cases. Figure 8 presents some failure cases for both VLN-BERT and Airbert. It reveals that current models still struggle to deal with relationships such as "*between*" (Figure 8a), or directional instructions such as "*on the left*" (Figure 8b). Similar failures are also highlighted by Table 5 of the main paper where we show that models fail to choose the correct instruction when a direction keyword (left/right) is switched.



(a) **R2R ✔:** Walk up the stairs and take a right. Walk into the bedroom and take a left . Take another left at the night stand and walk out of the bedroom. Wait by the toilet in the second door on the right.

(b) **R2R ✔:** Go straight past the table and chairs then turn left and continue to go past the table and chairs. Wait near the white antique furniture with the two chairs on on each side.

(c) **REVERIE ✔:** Walk past the pool table and towards the TV on the far side of the room and grab the coffee table that is located in front of the couch



(d) **REVERIE ✔:** Please go to the pantry room with the two large freezers and kitchen appliances on the large table and reset the flipped breaker in the breaker panel box to the right of the freezers

Figure 6: The Airbert model outperforms VLN-BERT to recognize rare or even unseen objects in training set. (a) Rare object "*night stand*"; (b) unseen object "*antique furniture*"; (c) rare object "*pool table*"; and (d) unseen object "*freezer*".

C.3. Few-shot Learning on VLN

As mentioned in the main paper, we present complete results for the few-shot learning evaluation, along with standard deviations in Table 2. While the performance on the seen validation houses fluctuates a lot (also due to changing the environment in the seen validation set), unseen validation is very stable. Recall that VLN-BERT achieves an unseen validation performance of 27% and 37% with 1 and 6 training environments respectively. On the other hand, Airbert achieves a superior 49.5% and 58% – an absolute improvement of ∼22% in both cases.

References

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018. 3

[2] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, 2018. 5

[3] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *3DV*, 2017. 8

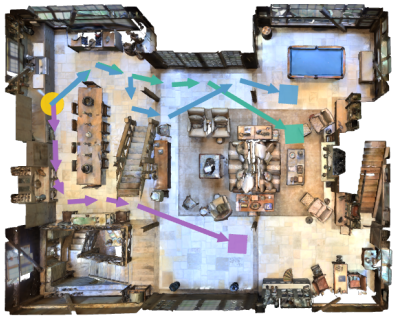
[4] Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-Follower models for vision-and-language navigation. In *NIPS*, 2018. 5

[5] Weituo Hao, Chunyuan Li, Xiujun Li, Lawrence Carin, and Jianfeng Gao. Towards learning a generic agent for vision-and-language navigation via pre-training. In *CVPR*, 2020. 3, 4, 5

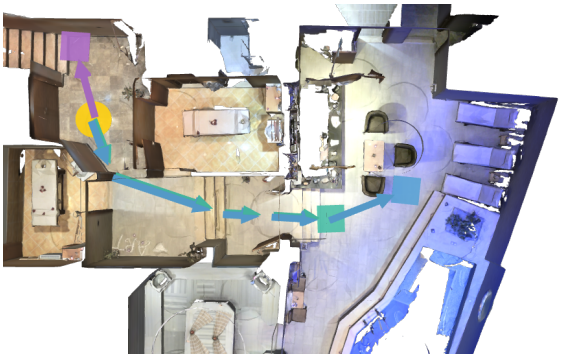
[6] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. A recurrent vision-and-language BERT for navigation. *arXiv preprint arXiv:2011.13922*, 2021. 3, 4, 5

[7] Xiujun Li, Chunyuan Li, Qiaolin Xia, Yonatan Bisk, Asli Celikyilmaz, Jianfeng Gao, Noah Smith, and Yejin Choi. Robust navigation with language pretraining and stochastic sampling. *EMNLP*, 2019. 5

(a) **R2R ✔:** Walk from dining room to living room turning slightly right before lounge chairs, walk straight following cabinet. Turn slight right and stop at stairs.



(b) **R2R ✔:** Walk on into the kitchen and turn to the right. Walk past the staircase, behind the chairs. Walk to the right of the pillar. Stop and wait by the footstool.



(c) **R2R ✔:** Walk out of the hallway and turn left. Walk down the steps and through the wooden doors. Walk down the steps and stop.

(d) **R2R ✔:** Go straight passed the coffee table turn left and go through the left door to the stairs. Stop in front of the stairs.

Figure 7: Examples in similar environments and instructions to the training set. The improvements of Airbert model can be contributed to the shuffling loss in pretraining.

# Env.	Traj.	Val Seen SR					Val Unseen SR				
		PL	NE	SPL	OSR	SR	PL	NE	SPL	OSR	SR
1	Rand	10.97	5.36	0.44	63.74	47.87 $\pm$0.03	10.84	4.86	0.51	68.46	54.48 $\pm$0.04
6	Rand	9.84	5.49	0.47	65.93	50.00 $\pm$0.02	9.55	4.55	0.55	70.89	57.97 $\pm$0.01
61	Rand	10.91	4.87	0.60	76.23	64.24	9.50	3.70	0.62	76.24	65.60
61	[11]	10.59	3.21	0.69	80.71	73.85	10.03	3.24	0.63	78.45	68.67

Table 2: Performance of Airbert on R2R few-shot evaluation. During training, only a subset of the Matterport [3] environments are accessible.

[8] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*, 2020. 3

[9] Arjun Majumdar, Ayush Shrivastava, Stefan Lee, Peter Anderson, Devi Parikh, and Dhruv Batra. Improving vision-and-language navigation with image-text pairs from the web. In *ECCV*, 2020. 3, 5

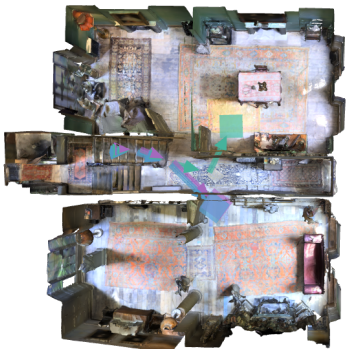
[10] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*, 2019. 3

[11] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. In *NAACL*, 2019. 5, 8

[12] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. *arXiv preprint arXiv:1506.03134*, 2015. 3

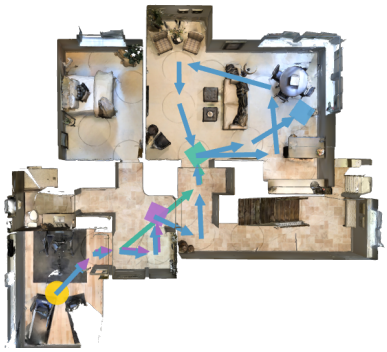
[13] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *PAMI*, 40:1452–1464, 2017. 1, 3

(a) **R2R ✘:** Walk between the two kitchen islands and then turn right. Pass through the stone archway and stop just after you pass through it. Wait there.



(b) **R2R ✘:** Exit the bathroom and go down the stairs. Enter the last doorway on the left and stop just before stepping on the rug.

(c) **REVERIE ✘:** go to level 3 bathroom in the first bedroom left of the stairs and grab the mirror on the wall



(d) **REVERIE ✘:** Go to the lounge on this level and polish the black leather armchair in the corner

Figure 8: Failure cases for both VLN-BERT and Airbert models.